

CLAIMS

1 1. A method for defragmenting data blocks on disks of a computer configured to imple-
2 ment a file system that logically organizes the blocks as a file on the disks, the file further
3 including indirect blocks having pointers that reference the data blocks, the method com-
4 prising the steps of:

5 determining a current layout of a range of pointers contained in each indirect
6 block of the file;

7 calculating a number of operations needed to retrieve the data blocks referenced
8 by the pointers from the disks to a memory of the computer;

9 estimating a potential new layout based on an average fullness of the file system;
10 and

11 relocating the data blocks on the disks if the potential new layout improves the
12 current layout.

1 2. The method of Claim 1 wherein the step of relocating comprises the step of relocating
2 the data blocks if there is sufficient free space on the disks.

1 3. The method of Claim 2 wherein the step of relocating comprises the steps of loading
2 the data blocks into the memory of the computer and dirtying the data blocks.

1 4. The method of Claim 3 wherein the step of relocating further comprises the steps of:
2 searching a predetermined distance of a first disk for free blocks; and
3 filling those free blocks with the dirtied data blocks.

1 5. The method of Claim 4 wherein the step of relocating further comprises the steps of:
2 jumping to a second disk;
3 searching the predetermined distance of the second disk for additional free blocks;
4 and
5 filling those additional free blocks with the dirtied data blocks.

1 6. The method of Claim 5 wherein the step of relocating further comprises the step of
2 repeating the steps of jumping, searching and filling until all data blocks of the file have
3 been relocated.

1 7. The method of Claim 5 wherein the predetermined distance is 32 data blocks.

1 8. A system adapted to defragment data blocks on disks of a computer configured to im-
2 plement a file system that logically organizes the blocks as a file on the disks, the file fur-
3 ther including indirect blocks having pointers that reference the data blocks, the system
4 comprising:

5 a processor coupled to the disks;
6 a memory coupled to the processor and having locations addressable by the proc-
7 essor; and

8 a storage operating system resident in the memory locations and executed by the
9 processor to invoke storage operations in support of the file system, the storage operating
10 system including a scanner adapted to (i) determine a current layout of a range of pointers
11 contained in each indirect block of the file, (ii) calculate a number of operations needed
12 to retrieve the data blocks referencing the pointers from the disks to the memory and (iii)
13 estimate a potential new layout based on an average fullness of the file system, the stor-
14 age operating system further including a write allocator adapted to relocate the data
15 blocks on the disks if the potential new layout improves fragmentation of the current lay-
16 out.

1 9. The system of Claim 8 wherein the file system is a write anywhere file system.

1 10. The system of Claim 9 wherein the scanner comprises a defragmentation process.

1 11. The system of Claim 10 wherein the range of pointers is a number of data blocks
2 referenced by the pointers of an indirect block.

1 12. The system of Claim 11 wherein the number of data blocks referenced by the point-
2 ers of an indirect block is 1024.

1 13. The system of Claim 8 wherein the memory is a buffer cache.

1 14. A method for defragmenting data blocks on disks of a computer configured to im-
2 plement a file system that logically organizes the blocks as a file on the disks, the file fur-
3 ther including indirect blocks having pointers that reference the data blocks, the method
4 comprising the steps of:

5 locating a beginning of the file;
6 selecting a range of data blocks to defragment;
7 attempting defragmentation of the range of data blocks; and
8 repeating the steps of selecting and attempting until defragmentation of all ranges
9 of data blocks within the file has been attempted.

1 15. The method of Claim 14 wherein the range of data blocks is a number of blocks ref-
2 erenced by an indirect block.

1 16. The method of Claim 15 wherein the number of blocks is 1024 blocks.

1 17. The method of Claim 14 wherein the step of attempting comprises the steps of:
2 reading pointers for the range of data blocks;
3 measuring an existing layout of the range of data blocks;
4 estimating a new layout of the range of data blocks;
5 determining whether the new layout is better than the existing layout;
6 if the new layout is better, determining whether there is sufficient free space in the
7 file system to relocate the data blocks;
8 if there is sufficient free space, reading the data blocks into a buffer cache and
9 dirtying the data blocks; and
10 rewriting the dirtied data blocks to new locations on the disks.

1 18. The method of Claim 17 wherein the step of measuring comprises the step of calcu-
2 lating a number of write_alloc_chunks needed to cover the data blocks in the range.

1 19. The method of Claim 18 wherein the number of write_alloc_chunks ranges from a
2 number of data blocks in the range and a number of data blocks in the range divided by a
3 write_alloc_chunk.

1 20. The method of Claim 18 wherein the step of estimating comprises the step of esti-
2 mating the new layout using an average fullness of the file system.

1 21. The method of Claim 18 wherein the step of determining whether the new layout is
2 better than the existing layout comprises the step of comparing existing and estimated
3 numbers of write_alloc_chunks.

1 22. Apparatus for defragmenting data blocks on disks of a computer configured to im-
2 plement a file system that logically organizes the blocks as a file on the disks, the file fur-
3 ther including indirect blocks having pointers that reference the data blocks, the apparatus
4 comprising:

5 means for determining a current layout of a range of pointers contained in each
6 indirect block of the file;

7 means for calculating a number of operations needed to retrieve the data blocks
8 referenced by the pointers from the disks to a memory of the computer;

9 means for estimating a potential new layout based on an average fullness of the
10 file system; and

11 means for relocating the data blocks on the disks if the potential new layout im-
12 proves the current layout.

1 23. The apparatus of Claim 22 wherein the means for relocating comprises:

2 means for searching a predetermined distance of the disks for free blocks; and

3 means for filling the free blocks with the data blocks.

1 24. A computer readable medium containing executable program instructions for de-
2 fragmenting data blocks on disks of a computer configured to implement a file system
3 that logically organizes the blocks as a file on the disks, the file further including indirect
4 blocks having pointers that reference the data blocks, the executable program instructions
5 comprising program instructions for:

6 determining a current layout of a range of pointers contained in each indirect
7 block of the file;

8 calculating a number of operations needed to retrieve the data blocks referenced
9 by the pointers from the disks to a memory of the computer;

10 estimating a potential new layout based on an average fullness of the file system;
11 and

12 relocating the data blocks on the disks if the potential new layout improves the
13 current layout.

1 25. The computer readable medium of Claim 24 wherein the program instruction for re-
2 locating comprises program instructions for:

3 searching a predetermined distance of the disks for free blocks; and
4 filling the free blocks with the data blocks.